The Attentive Perceptron

Raphael A. Pelossof
Columbia University
pelossof@cs.columbia.edu

Zhiliang Ying
Columbia University
zying@stat.columbia.edu
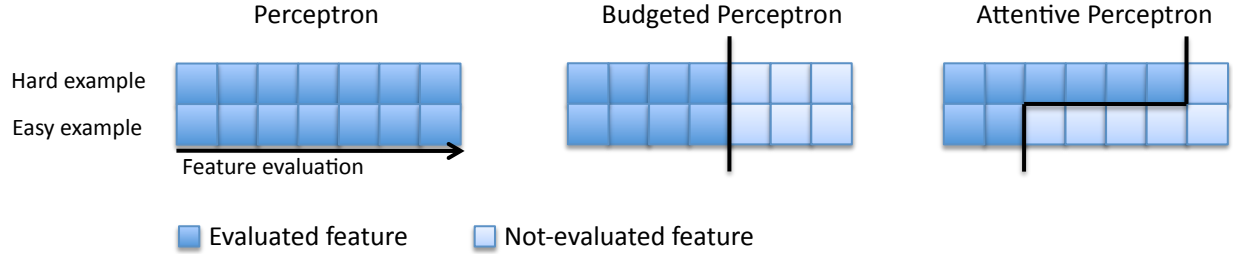
Figure 1: *The Attentive Perceptron adaptively allocates computational effort according to how hard an example is to classify. While the traditional Perceptron evaluates all the features for all the examples, a Budgeted Perceptron can only evaluate a constant number of features which is limited by the hard budget. From a budgeted learning point of view, the Attentive Perceptron adaptively allocates computation while maintaining an average budget. Therefore easily classifiable examples are filtered after having evaluated a few of their features, whereas hard to classify examples have the majority of their features evaluated.*

Abstract

We propose a focus of attention mechanism to speed up the Perceptron algorithm. Focus of attention speeds up the Perceptron algorithm by lowering the number of features evaluated throughout training and prediction. Whereas the traditional Perceptron evaluates all the features of each example, the Attentive Perceptron evaluates less features for easy to classify examples, thereby achieving significant speedups and small losses in prediction accuracy. Focus of attention allows the Attentive Perceptron to stop the evaluation of features at any interim point and filter the example. This creates an attentive filter which concentrates computation at examples that are hard to classify, and quickly filters examples that are easy to classify.

1 Introduction

Many Online Algorithms base their model update on the margin of each example. Passive online algorithms, such as Rosenblatt's Perceptron [7] and Crammer et al's online passive-aggressive algorithms [3], update the algorithm's model only if the value of the margin falls below a defined threshold. These algorithms fully evaluate the margin for each example, even if the model is not to be updated!

The running time of these algorithms is linear either in the number of features, or in the dimensionality of the input space. Contemporary models may have thousands of features making running time daunting. The budgeted learning community addresses this problem by putting a budget on the number of features a classifier can evaluate while learning and while making predictions. Our work stems from the theoretical framework suggested by Ben David and Dichterman [1], and is closely related to recent work by Cesa-Bianchi et al. [2] as well as Reyzin [6].

We differ by the fact that we do not impose a hard budget constraint on the number of features, but rather look at the probability of making decision errors. Decision error are errors that occur when the algorithm stops the feature evaluation process, predicts its outcome, and is wrong. This work extends on previous work by Pelossof et al. [5].

We propose a new method for early stopping the computation of feature evaluations for uninformative examples by connecting the Perceptron algorithm to sequential statistical tests [8, 4] (Figure 1.) This connection results in a general method that makes margin based learning algorithms attentive, which means that they have the ability to quickly filter uninformative examples.

2 The Attentive Perceptron

The margin of each example is computed as a weighted sum of feature evaluations. Informative examples are misclassified examples, which force the Perceptron to preform a model update, whereas uninformative examples are correctly classified and therefore ignored by the perceptron.

We break up the feature evaluation for every example in the stream. The breakup of every example allows the Attentive Perceptron to make a decision after the evaluation of each feature about whether the feature evaluation should continue or be stopped. This decision making process allows us to stop the evaluation of features early on examples with a large partial margin after having evaluated only a few features. For example, examples with a large partial margin are unlikely to have a negative full margin. Therefore, rejecting these examples early achieves large savings in computation.

We define the mathematical setup to derive the stopping conditions for margin evaluation. Let $X_1, ..., X_n$ be weakly dependent random variables. Let a partial sum be defined by $S_i = X_1 + ... + X_i$ and the remainder sum by $S_{in} = S_n - S_i$. The expectation of a sum is denoted by ES_i and its standard deviation by $std(S_i)$.

The Perceptron compares the margin (a sum) to a threshold, and updates its model if the margin of the example is negative. We formulate the equivalent sequential decision making process, and drive constant stopping thresholds τ. These thresholds will essentially tell us when it's highly unlikely for the margin to end below the desired importance threshold θ.

The stopping thresholds are derived by requiring that the joint distribution of stopping (and predicting $S_n > \theta$) while the actual full sum satisfies $S_n < \theta$ is less than a required error rate δ

$$P(S_n < \theta, predict \ S_n > \theta) = P(S_n < \theta, S_i > \tau) \leq \delta.$$

We bound the probability of making a decision error

$$
\begin{aligned}
P(S_n < \theta, S_i > \tau) &\lesssim P(S_n < \theta, S_i = \tau) \\
&= P(S_n - ES_n < \theta - ES_n, S_i = \tau) \\
&= P(S_n - ES_n < 2\tau - (\theta - ES_n)) \quad (1) \\
&= P\left(\frac{S_n - ES_n}{std(S_n)} < \frac{2\tau - \theta + ES_n}{std(S_n)}\right). \quad (2)
\end{aligned}
$$

Equation 1 is derived by applying the reflection principle, and equation 2 is its standardization.

Since we assume that $X_1, ..., X_n$ are weakly independent, the sum $S_n = X_1 + ... + X_n$ is approximately normally distributed by the Central Limit Theorem. By standardizing S_n we upper bound the probability of making a decision error with the inverse normal cumulative distribution function Φ^{-1}. Therefore, requiring that the probability of making a decision error be less than δ we get the following equality from equation 2

$$\frac{2\tau - \theta + ES_n}{std(S_n)} = \Phi^{-1}(1 - \delta). \quad (3)$$

The quantities ES_n and $std(S_n)$ can be approximated using the empirical data.

Finally, by solving for the stopping threshold τ we get from equation 3

$$\tau = \frac{1}{2}\left(\theta - ES_n + std(S_n)\Phi^{-1}(1 - \delta)\right). \quad (4)$$

Therefore, examples with partial margin calculations S_i that hit this boundary should be filtered and with probability at least $1 - \delta$ determined that their full margin satisfies $S_n > \theta$.

In summary, we presented a simple test to speed up the Perceptron algorithm by quickly filtering unimportant examples without fully evaluating their features. This results in an algorithm which typically focuses on examples by the decision boundary - the Attentive Perceptron.

References

[1] Shai Ben-David and Eli Dichterman, *Learning with restricted focus of attention*, Journal of Computer and System Sciences **56** (1998), no. 3, 277 – 298.

[2] Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir, *Efficient learning with partially observed attributes*, ICML, 2010.

[3] Koby Crammer Crammer, Ofer Dekel, Joseph Keshet, and Yoram Singer, *Online passive-aggressive algorithms*, JMLR **7** (2006), 551–585.

[4] K.K. Gordon Lan, Richard Simon, and Max Halperin, *Stochastically curtailed tests in long-term clinical trials*, Sequential Analysis **1** (1982), no. 3, 207–219.

[5] Raphael Pelossof, Michael Jones, and Zhiliang Ying, *Curtailed online boosting*, ICML, BL workshop, 2010.

[6] Lev Reyzin, *Boosting on a feature budget*, ICML, BL workshop, 2010.

[7] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain*, Psychological Review **65** (1958), no. 6, 386–408.

[8] A. Wald, *Sequential tests of statistical hypotheses*, The Annals of Mathematical Statistics **16** (1945), no. 2.